Function approximation

Laverty

10/22/2019

Function interpolation and approximation

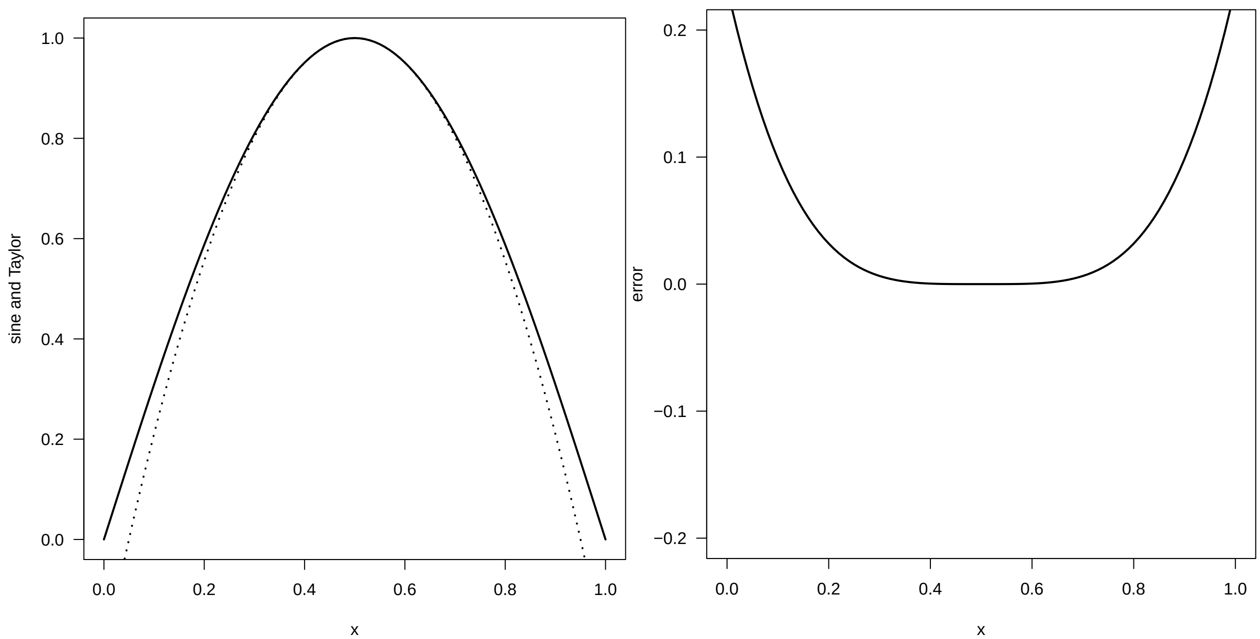
Below we discuss $f(x) = \sin(\pi x)$ on $[0, 1]$ and three interpolation/approximation schemes.

A reasonable starting point is the polynomial given by the second order Taylor polynomial

$$T_2(2) = -0.233701 + x\,(4.9348 - 4.9348x)$$

This is

```
par(mar=c(4.1, 4.1, 1.1, 1.1))
f <- function(x) sin(pi*x)
plot(f, xlim=c(0, 1), lwd=2, las=1, ylab="sine and Taylor", xlab="x")
T2 <- function(x) -0.233701 + x*(4.9348 - 4.9348*x)
plot(T2, xlim=c(0, 1), lty=3, lwd=2, add=T)
plot(function(x) f(x)-T2(x), xlim=c(0, 1), ylim=c(-0.2, 0.2), lwd=2, las=1, ylab="error")
```

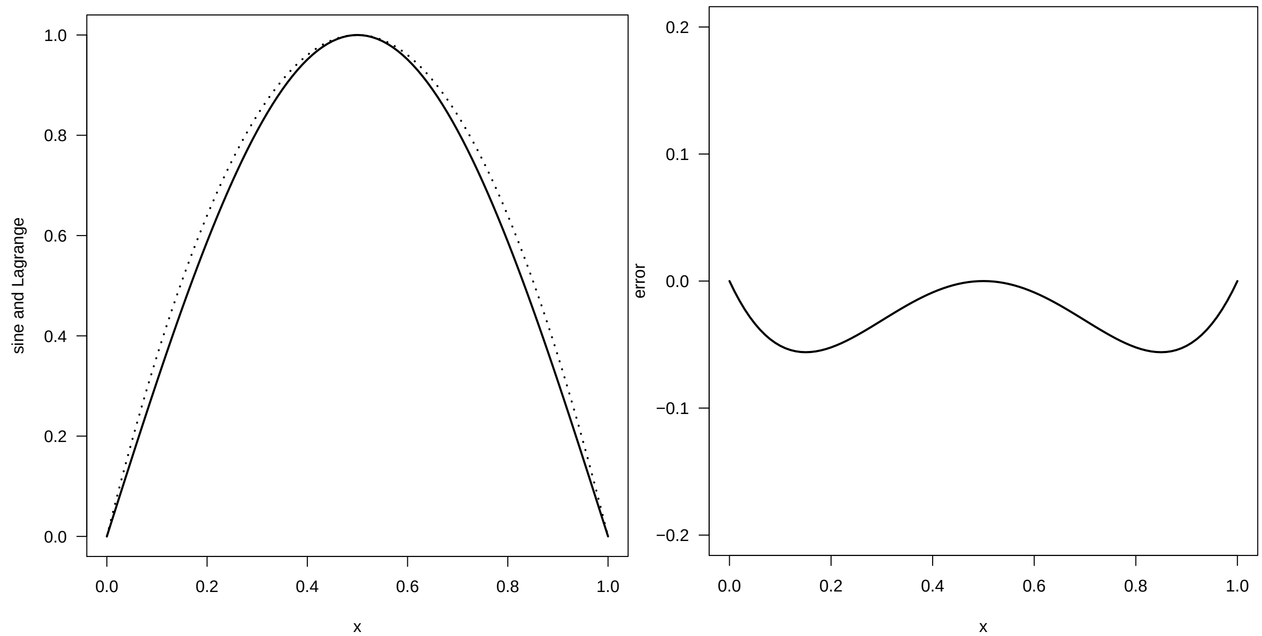


A reasonable next step is the polynomial given by the Lagrange interpolating polynomial

$$P_2(2) = x(4 - 4x)$$

This is

```
par(mar=c(4.1, 4.1, 1.1, 1.1))
f <- function(x) sin(pi*x)
plot(f, xlim=c(0, 1), lwd=2, las=1, ylab="sine and Lagrange", xlab="x")
P2 <- function(x) x*(4 - 4*x)
plot(P2, xlim=c(0, 1), lty=3, lwd=2, add=T)
plot(function(x) f(x)-P2(x), xlim=c(0, 1), ylim=c(-0.2, 0.2), lwd=2, las=1, ylab="error")
```

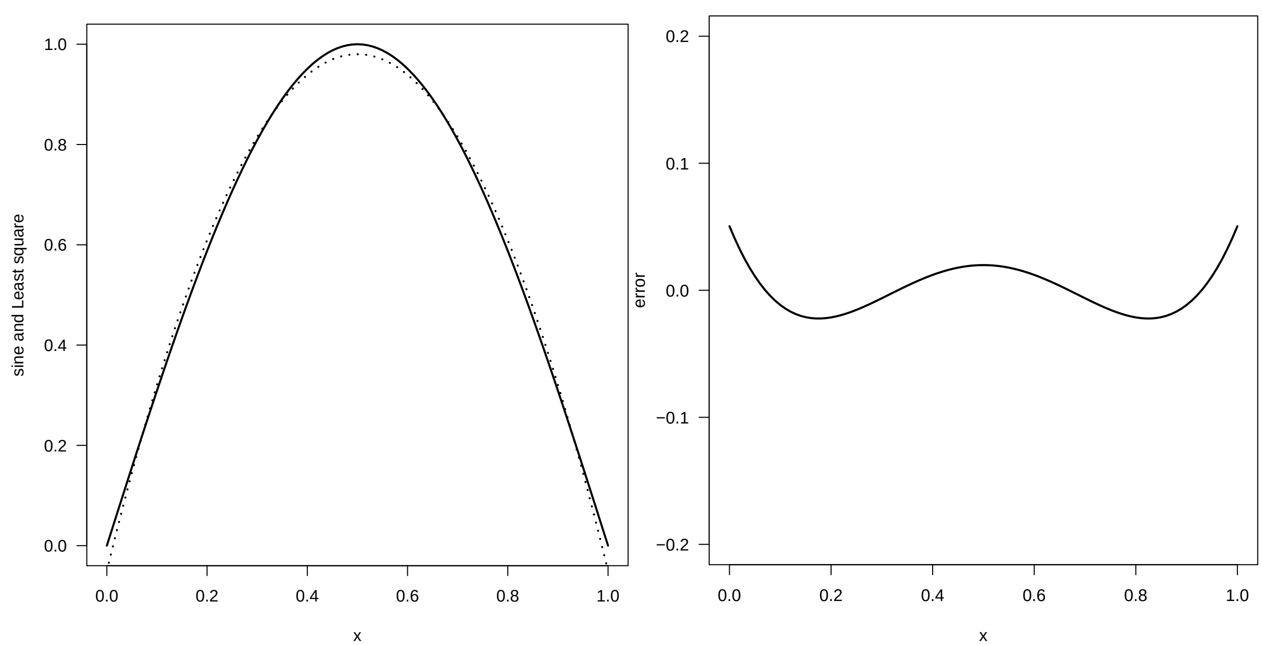


Applying the least squares approximation technique we find

$$L_2(2) = -0.0504655 + x(4.12251 - 4.12251x)$$

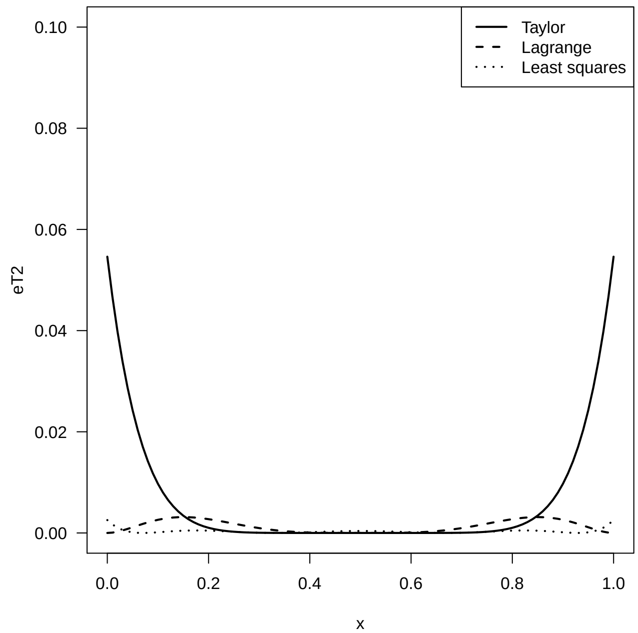
This is

```
par(mar=c(4.1, 4.1, 1.1, 1.1))
f <- function(x) sin(pi*x)
plot(f, xlim=c(0, 1), lwd=2, las=1, ylab="sine and Least square", xlab="x")
L2 <- function(x) -0.0504655 + x*(4.12251 - 4.12251*x)
plot(L2, xlim=c(0, 1), lty=3, lwd=2, add=T)
plot(function(x) f(x)-L2(x), xlim=c(0, 1), ylim=c(-0.2, 0.2),  lwd=2, las=1, ylab="error")
```



In terms of the errors we have.

```r
par(mfrow=c(1, 1), mar=c(4.1, 4.1, 1.1, 1.1))
eT2 <- function(x) (f(x) - T2(x))^2
eP2 <- function(x) (f(x) - P2(x))^2
eL2 <- function(x) (f(x) - L2(x))^2
plot(eT2, xlim=c(0, 1), ylim=c(-0.0, 0.1), lwd=2, lty=1, las=1)
plot(eP2, xlim=c(0, 1), lwd=2, lty=2, add=T)
plot(eL2, xlim=c(0, 1), lwd=2, lty=3, add=T)
legend("topright", c("Taylor", "Lagrange", "Least squares"), lwd=2, lty=c(1,2,3))
```



The squared errors are quite small in magnitude, but when integrated from $[0, 1]$ we have the results shown in the following table. Keep in mind that the Taylor and Lagrange polynomials were derived before the concept of least squares error was introduced - so perhaps this is an unfair way to measure them. By other metrics or in other circumstances, these are still useful approximations.

Method	Formula	$err_{poly} = \int_0^1 (f(x) - \text{poly.})^2 \, dx$
Exact	$f(x) = \sin(\pi x)$	
Taylor poly. $(x_0 = \frac{1}{2})$	$T_2(x) = -0.233701 + x\,(4.9348 - 4.9348x)$	0.00625361
Lagrange poly. $(x_i = 0, \frac{1}{2}, 1)$	$P_2(x) = x(4 - 4x)$	0.00128423
Least squares quadratic	$L_2(x) = -0.0504655 + x(4.12251 - 4.12251x)$	0.00029803

In terms of the squared error, the least squares formulation performs about 20 times better than the Taylor polynomial (better meaning $err_{L_2} \approx \frac{1}{20} err_{T_2}$) and about 4.3 times better than the Lagrange polynomial (better meaning $err_{L_2} \approx \frac{1}{4} err_{P_2}$). Lagrange itself is about 4.9 times better than the Taylor polynomial (better meaning $err_{P_2} \approx \frac{1}{5} err_{T_2}$).